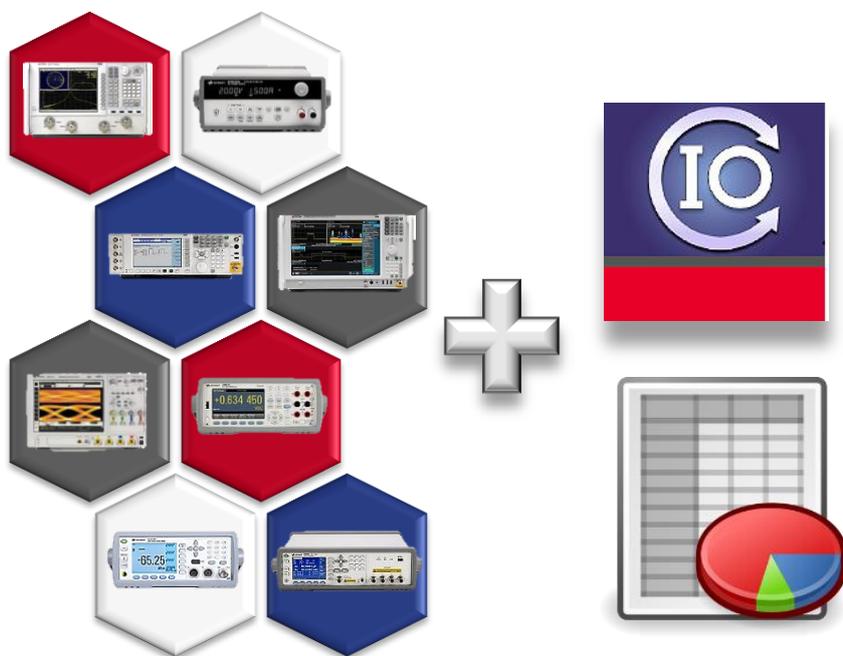
初心者向け
Excel / VBA による測定器制御プログラム　超入門編

測定器制御のための Excel / VBA の基礎知識

Application Note

目次

はじめに

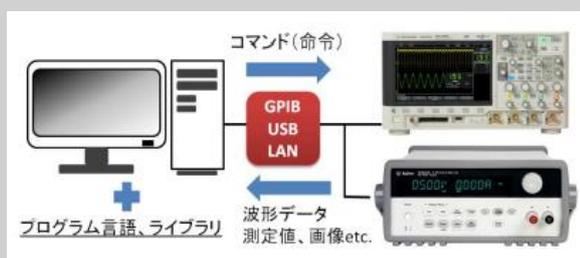
Excel / VBA を使用すると PC に接続されている測定器を制御し、測定値を Excel のセルに保存するプログラムを作成することができます。
今までにプログラムを作成した経験がないと、どのような作業が必要になるのか、何を調べればよいのか、ということが不明なため、プログラム作成をあきらめてしまうこともあるのではないでしょうか？

本アプリケーション・ノートはプログラムを作成した経験のないエンジニアの方を対象に、Excel / VBA で以下のプログラムの動作とプログラムを作成する実際の手順を説明します。

1. データに名前を付けて保存する、読み出す、演算を行う
2. 演算結果により、次に実行する内容を変更する
   1) 条件分岐処理
   2) 繰り返し処理

これにより、自動計測プログラム作成を開始するために必要な、基礎知識を理解することを目的とします。
プログラムを記述する言語は Visual Basic や Excel / VBA など多くの言語があります。
それぞれの言語には特徴があり、目的や用途によって選択できます。
この資料では Excel / VBA を使用して説明します。

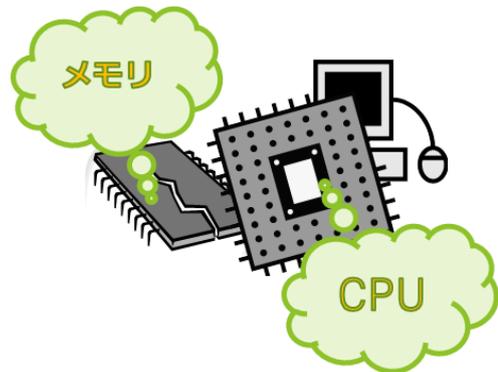
キーサイト・テクノロジーでは Excel VBA や Visual Basic .NET、Visual C#、Visual C++、Keysight VEE で測定器を制御するサンプル・プログラムやアプリケーション・ノートを公開、またトレーニングの実施も行っておりますので合わせてご参照ください。(この資料の最後のページをご参照ください。)

[自動制御トレーニング・コース]
http://www.keysight.co.jp/find/programing103

1. プログラムとは

プログラムとは『 PC に実行させる一連の命令を、あらかじめ順番に記述したもの』です。
PC には CPU とメモリがあり、CPU はメモリ上のプログラムを実行します。

プログラムで使用できる命令 (CPU で行う処理) には多くの種類がありますが、基本的に
1. 数値、文字などのデータに名前を付けてメモリに保存、およびメモリから読み出す
2. データの演算を行う
3. 演算結果により、次に実行する内容を変更する

を行います。

PC はプログラムを実行するときにメモリを使用しますが、メモリは CPU の要求に応じて、プログラムやデータに名前を付けて保存する、保存されたプログラムやデータを CPU の要求に応じて CPU に返す、という機能を持っています。

2. データに名前を付けて保存する、読み出す、演算を行う

二つの数値を加算する、というプログラムを Excel で実行させる場合、計算する値や計算した結果は PC のメモリに保存されていますが、ユーザーは計算する値を変更したり、計算結果を直接見ることはできません。

以下は Excel が実行できるプログラムの例です。

```
A = 5
B = 3
C = A + B
```

Excel で上記のプログラムを実行すると、Excel はプログラムを上から下に、順番に 1行ずつ実行します。

1. Excel は最初に 1 行目の A = 5 を実行します

→ Excel はコンピュータのメモリに、「A という名前を付けた箱 (変数 A) 」 を準備します
次に 「5 という数値」 を準備します
最後に 5 という数値を A という名前を付けた箱に 「保存」 します
「=」 はデータ (値) をメモリに保存する (代入) 命令です。

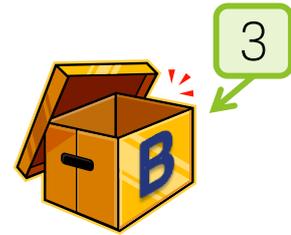
補足：変数

メモリ上の「名前の付いた箱」のことを、プログラム用語では『変数』と言います。
A という名前の付いた箱のことを 『A という変数』、『変数A』 等と言います。

補足：代入

「保存する」という動作のことを、プログラム用語では『代入する』と言います。
プログラム用語では「C = A + B」 は、『変数A と変数B を加算して変数C に代入する』と言います。

2.　次に 2 行目の B = 3　を実行します

→ 1. と同様に、コンピュータのメモリに、「B という名前を付けた箱」 を準備し、
次に「3 という数値」 を準備し、
B という名前を付けた箱に 3 という数値を 「保存」 します

3.　3 行目の C = A + B を実行します

→ メモリに 「C という名前を付けた箱」 を準備し、
「A の箱からデータを取り出し」 ここでは 『5』 を取出し、
「B の箱からデータを取り出し」 ここでは 『3』 を取出し、
「+ の計算」を行います
計算結果は 8 で、これを C という名前を付けた箱に 「保存」 します

数学では 『 = 』 という記号は「等しい」という意味ですが、この例の使用方法では左辺に指定されている名前の付いた箱に、右辺の計算結果を 「保存する」という意味です。
C = A + B のようなプログラムでは最初に A + B を実行し、実行結果を C に保存します。

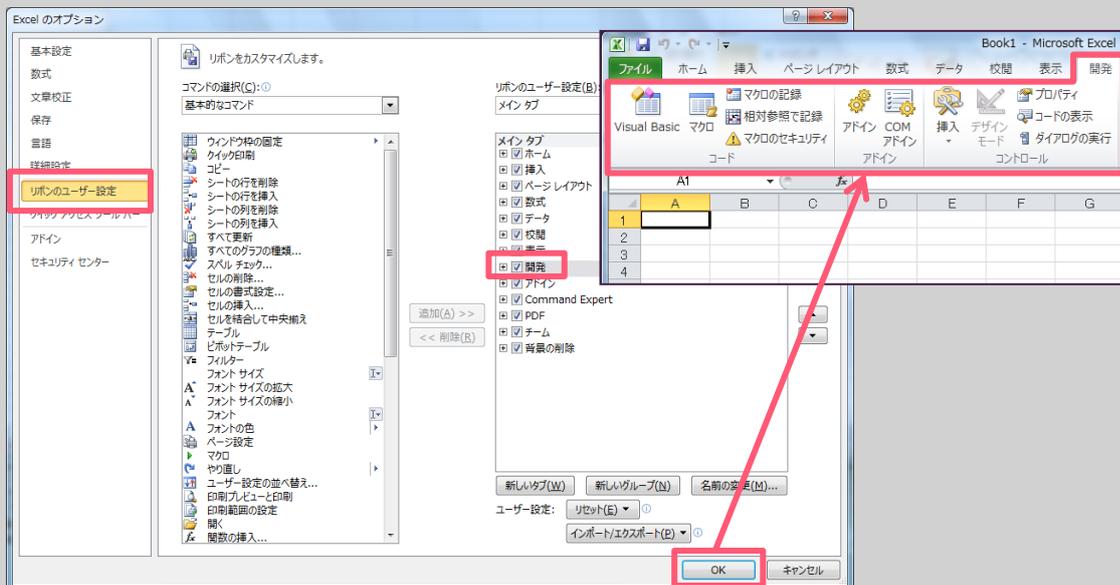
『 + 』 の記号は数学と同じで 「加算する」 という意味です。

例えば、セル A1 の数値と、セル A2 の数値を加算し、結果をセル A3 に表示する、という動作を行えばセル A1 に入力する数値と、セル A2 に入力する数値を変更するだけで、計算する数値を変更したり、結果を確認することが可能です。

次に Excel シート上のボタンを押すことで、セル A1 とセル A2 を加算し、セル A3 に表示するプログラムを作成します。

補足：開発タブの表示（Excel　2007 / 2010 / 2013 の場合）

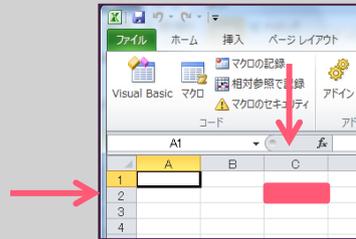
初めて Excel VBA でプログラムを作成する場合、Excel のリボンに『開発』タブを表示させrます。

1. Excel を起動します
2. Excel 2010 / 2013 の場合：
　　ファイル → Excel のオプション → リボンのユーザー設定 → 「開発」にチェック
　Excel 2007 の場合：
　　Office ボタン → Excel のオプション → 基本設定 → 「開発タブをリボンに表示する」にチェック

1. セルの内容をプログラムから使用する方法

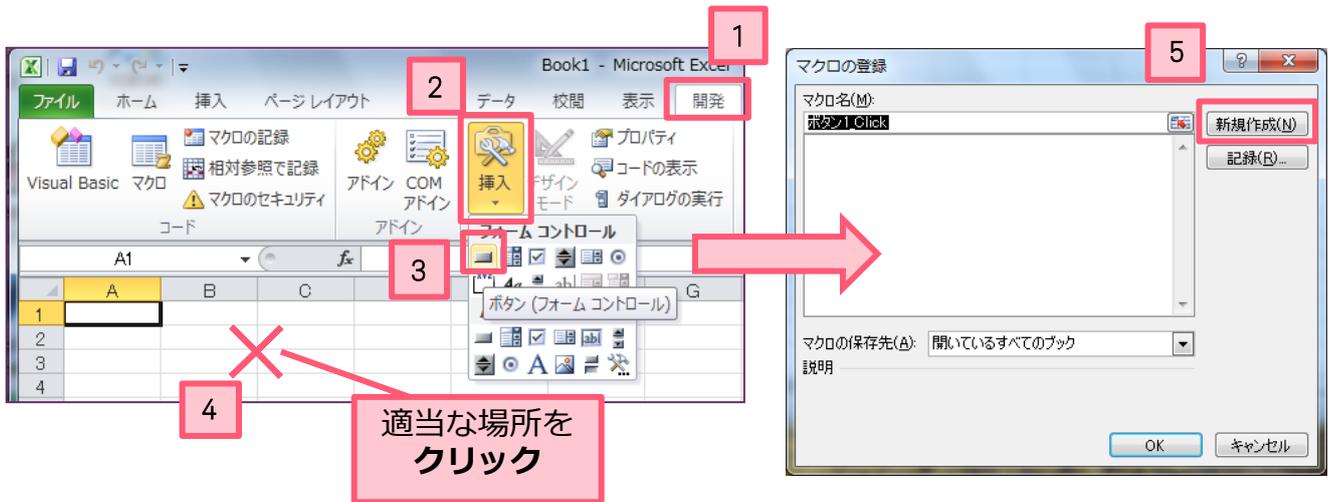
Excel には『 Cells 』というキーワードが
予め提供されています。
Excel のプログラムで、キーワード Cells を使用
することでシート上のセルを指定して、
　「セルに保存されているデータを取り出す」
　「セルにデータを保存する」
などの操作が可能です。

補足：Cells

このプログラムでは Cells というキーワード (予約語) を使用します。
Cells は Excel にあらかじめ提供されている機能です。
Cells (行, 列) でデータ (値) の代入、参照が可能です。

例：　Cells (2, 3)　→　2行3列　→　セル C2 を示します。

例えば、以下のプログラムは
　A = Cells (1, 1)　　‘ 変数 A にセル A1 のデータを代入
　B = Cells (2, 1)　　‘ 変数 B にセル A2 のデータを代入
　Cells (3, 1) = A + B　　‘ 変数 A のデータと変数 B の内容を加算して、結果をセル A3 に代入

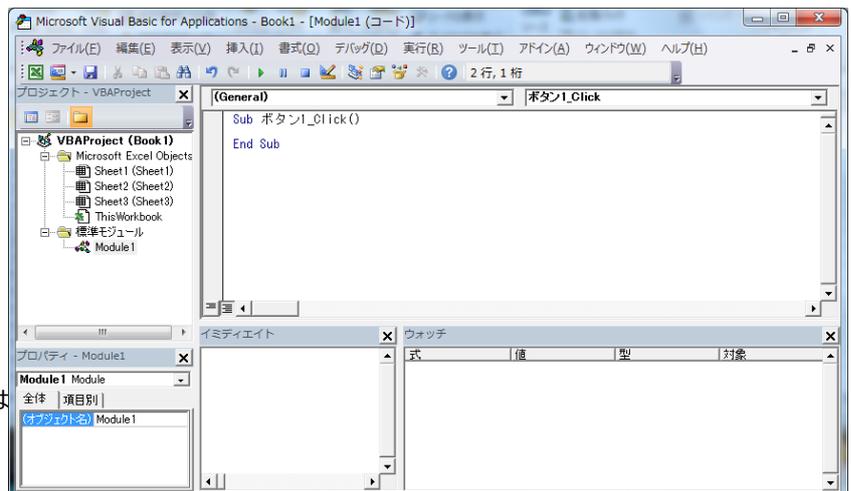
2. シートにボタンを配置して、ボタンが押されたらプログラムを実行する方法

開発タブを選択し、挿入 → フォームコントロール → ボタンを選択し、シート上の適当な場所にボタンを配置します。
表示されるダイアログの [マクロの登録] で、新規作成を選択します。

表示される Visual Basic Application Editor で、以下のように表示されます。

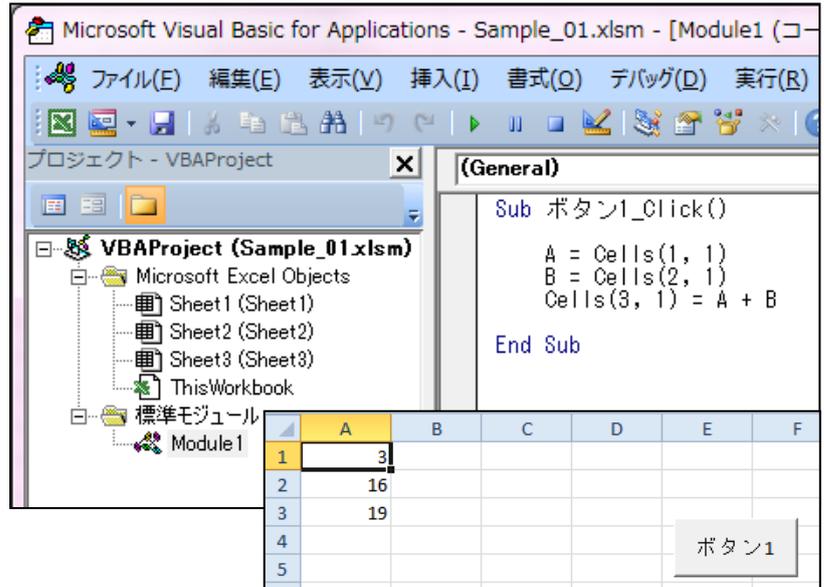
```
Sub ボタン1_Click()

End Sub
```

これはシート上のボタンをクリックする
ことで Excel が実行するプログラムを登録
する画面です。

Sub ボタン1_Click()　から

End Sub　までの間に記入したプログラムは
シート上のボタンをクリックすると実行
されます。

以下のように入力してください。

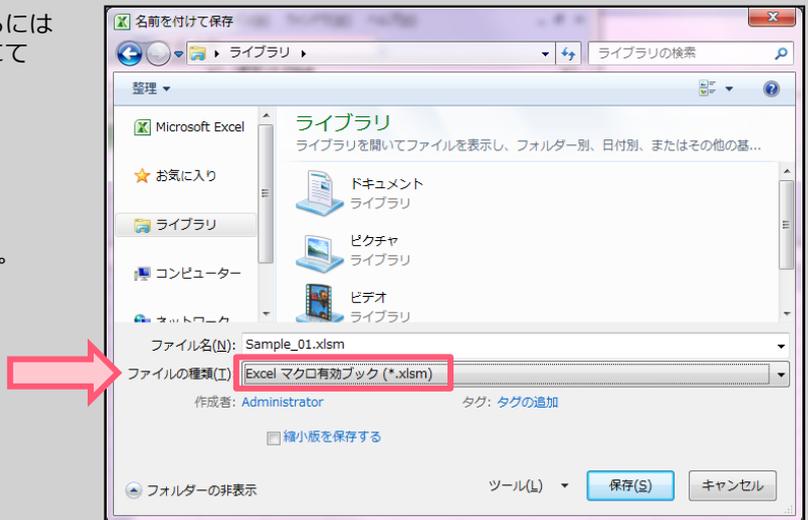
```
Sub ボタン1_Click()
    A = Cells (1, 1)
    B = Cells (2, 1)
    Cells (3, 1) = A + B
End Sub
```

Excel シートに戻り、セル A1、セル A2 に数字を入力し、[ボタン1] をクリックしてください。
セル A3 に合計が表示されます。

補足：VBA プログラムの保存方法 (Excel 2007 以降)

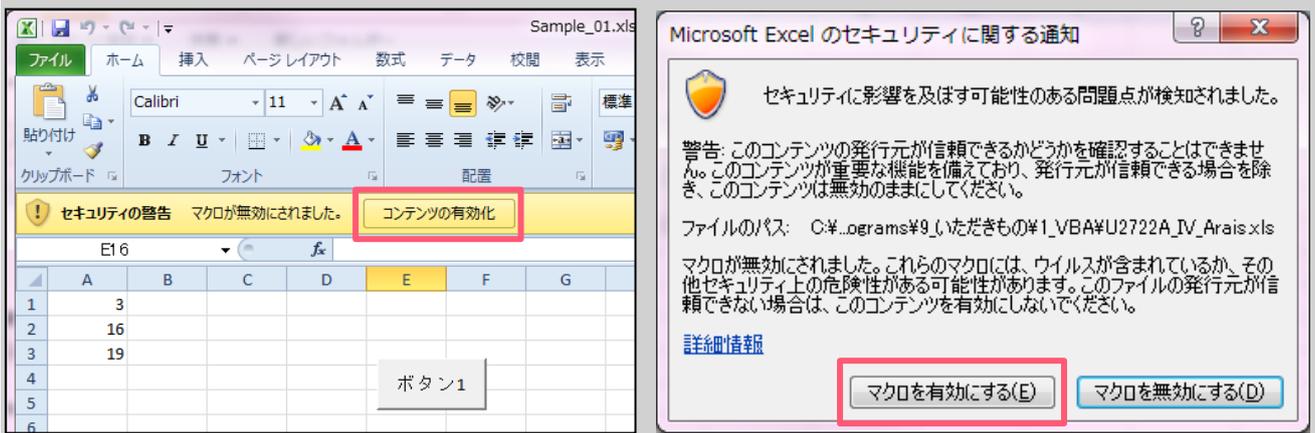
プログラムを含めてファイルとして保存するには
Excel の　ファイル → 名前を付けて保存　にて
ファイルの種類を

『 Excel マクロ有効ブック (*.xlsm) 』

にする必要があります。

(デフォルトは　Excel ブック (*.xlsx)　です。
.xlsx の拡張子のファイルで保存した場合、
プログラムは保存されません。)

補足：保存した .xlsm ファイルを開いてプログラムを実行する方法

作成した .xlsm ファイルを開くとセキュリティの警告が出ることがあります。
信頼できるファイルの場合は『 有効化 』するとプログラムが実行できます。
信頼できないファイルの場合はコンテンツを有効にしないでください。

プログラムを実行できない場合は一度ファイルを閉じ、再度開くと [コンテンツの有効化] ボタンが出てきます。

3. 演算結果により、次に実行する内容を変更する（条件分岐処理、繰り返し処理）

CPU の演算結果により、次に実行する内容を変更する処理について説明します。

たとえば演算の結果が 100 より大きければ “Pass”、100 以下であれば “Fail” とセルに表示するプログラムを作成するとします。
この場合、条件分岐処理が必要です。
また、測定を 100 回実行する場合、繰り返し処理を行うことで効率よくプログラムが可能です。

先に説明した上から下に1行ずつ実行するプログラムに、条件分岐処理と繰り返し処理を組み合わせることにより、より実用的なプログラムを作成することができます。
ここでは具体的な例を示して説明します。

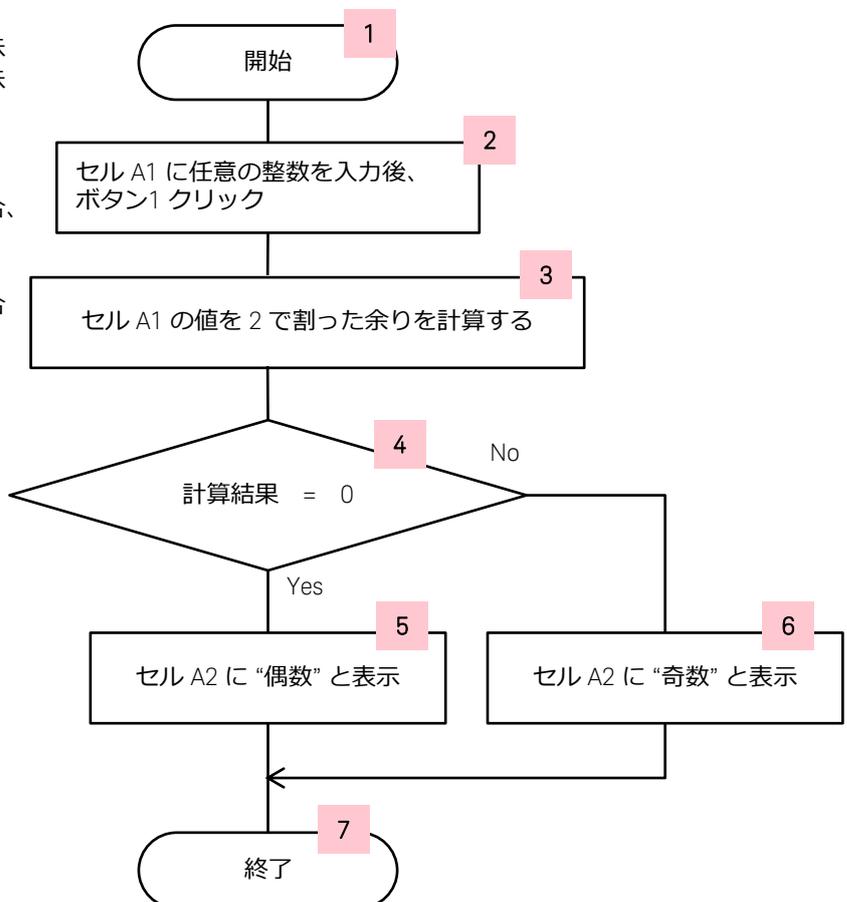
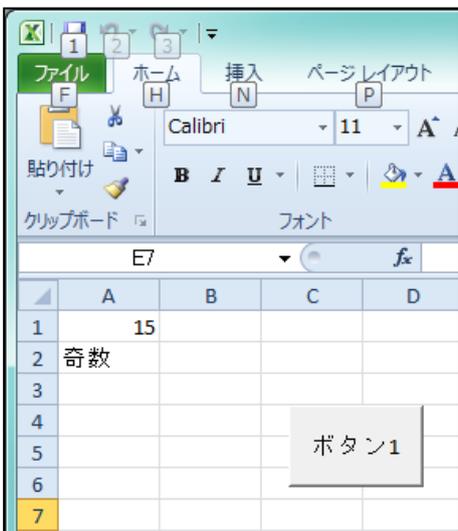
1) 条件分岐処理　If Then / End If

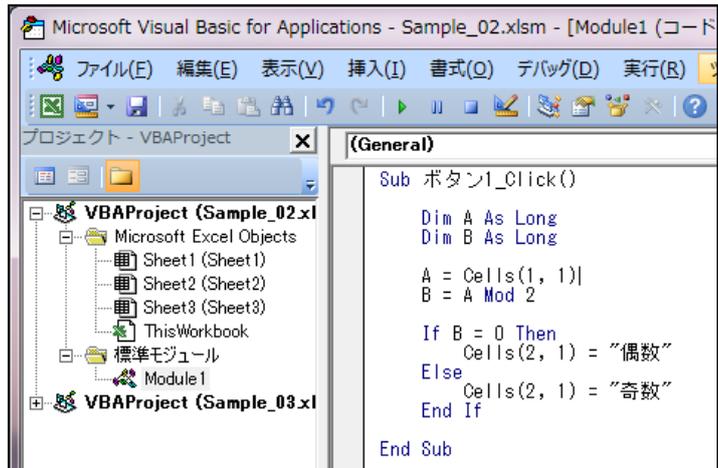
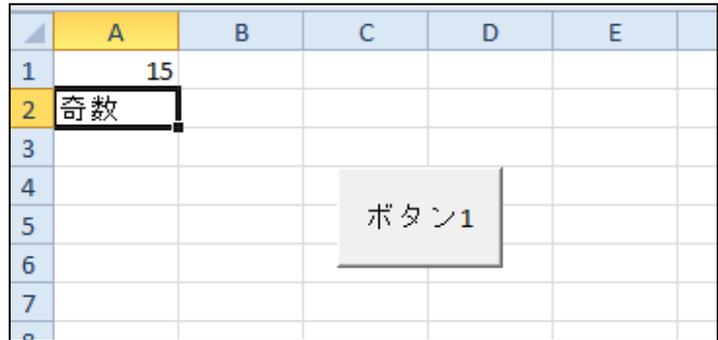
演算結果によって、異なる処理を行うことができます。
例えば、製造した製品を測定し、測定結果によって良品と不良品を判別する場合に条件分岐処理を使用します。

以下の例はセル A1 に入力した任意の整数が偶数か奇数かをセル A2 に結果と して表示します。

1. 開始
2. Excel のセル A1 に任意の整数を入力し、[ボタン1] を押す
3. セル A1 の値の 2 の剰余を演算する
4. 剰余が「0」かどうかを判断
5. 剰余が「0」ならセル A2 に ”偶数” と表示
6. 剰余が「1」ならセル A2 に “奇数” と表示
7. 終了

実行例：　セル A1 に 「15」 を入力した場合、
　1 → 2 → 3 → 4 → 6 → 7　を実行

実行例：　セル A1 に 「16」 を入力した場合
　1 → 2 → 3 → 4 → 5 → 7　を実行

このプログラムは以下のプログラムで実現できます。

```
Sub ボタン1_Click()

  Dim A As Long
  Dim B As Long

  A = Cells(1, 1)
  B = A Mod 2

  If B = 0 Then
    Cells(2, 1) = "偶数"
  Else
    Cells(2, 1) = "奇数"
  End If

End Sub
```

補足：Mod 演算子 (剰余)

Excel VBA で、剰余を取得するには Mod 演算子を使用します。

例：　14 Mod 3 の演算結果は 「2」
　　　→ 14 ÷ 3 = 4 ・・・ 2

このプログラムでは、If B = 0 Then から End If が条件分岐を行っている部分です。

B (セル A1 の 2 の剰余) が 「0」 と等しければ、Then から Else までの Cells (2, 1) = “偶数” が実行され、セル A2 に「偶数」と表示されます。
B (セル A1 の 2 の剰余) が 「0」 と等しくなければ、Else から End If までの Cells (2, 1) = “奇数” が実行され、セル A2 に「奇数」と表示されます。

VBA では『 = 』は前述した「代入」以外に数学と同じ「等しい」という意味で使用することがあります。
If 文中の『 = 』は、条件式で左辺と右辺が『等しい』という意味で使用します。
If B = 0 Then というコードは「もし、B が 0 と等しければ」という意味です。

補足：文字列のデータは “ ” で囲む

上記のプログラムプログラムでセル A2 に代入するデータ (偶数、奇数) の両端を『” (ダブルクォーテーション)』で囲っています。Excel VBA では文字列のデータはダブルクォーテーションで囲むというルールがあります。

補足：コメント

プログラムをわかりやすくするために、プログラム上にコメントを記述することができます。
コメント部分はプログラムを実行しても動作しません。
コメントを記述するには『 ’ (アポストロフィー、シングルクォーテーション)』の後に記述します。

例： If A < C Then　‘ 条件分岐処理 A < C の場合
　　上記の場合、「 If A < C Then 」 は実行されますが、「 ‘ 条件分岐処理 A < C の場合」 は実行されません。

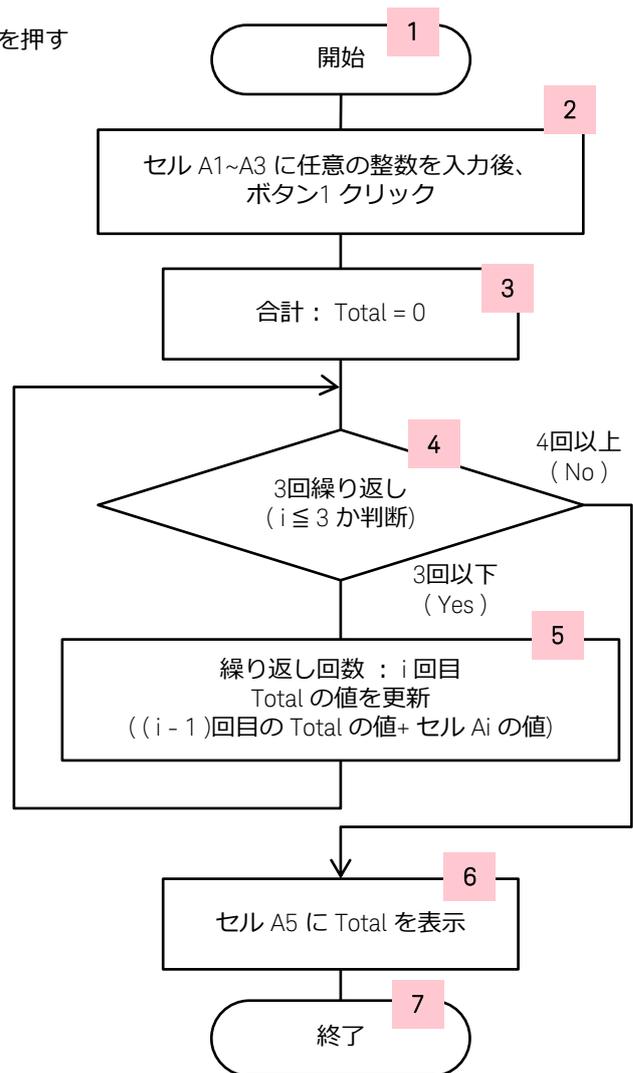
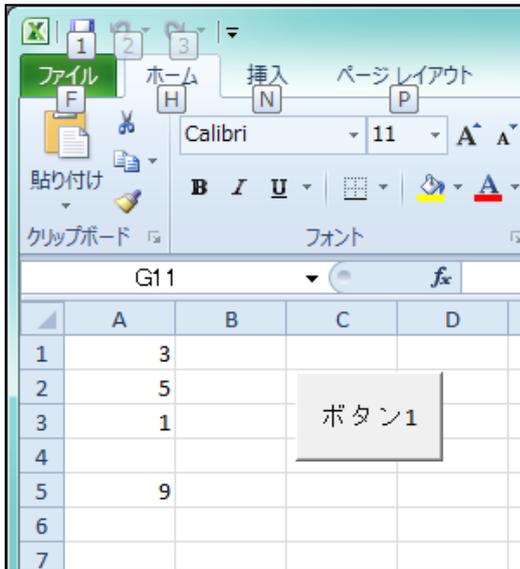
2) 繰り返し処理 (ループ)　For Next

同じ処理を繰り返します。条件分岐処理の一種です。

以下の例はセル A1 ～ A3 の値を加算し、結果をセル A5 に表示します。

ここでは繰り返しの回数を変数 i を使って表記します。　　例：i =1 の場合　→　i 回目 = 1回目、セル Ai = セル A1
またセルの値を加算するために変数 Total を使用します。
変数 Total の初期値は 「0」 を代入し、繰り返し処理でセル A1 の値から順番に加算していきます。

1. 開始
2. Excel のセル A1 ～ A3 に任意の整数を入力し、[ボタン1] を押す
3. 変数 Total に初期値 「0」 を代入
4. 繰り返し回数が 3回 以下か判断 (条件分岐処理)
5. セル Ai の値を変数 Total に加算する
6. セル A5 に変数 Total の値を表示
7. 終了

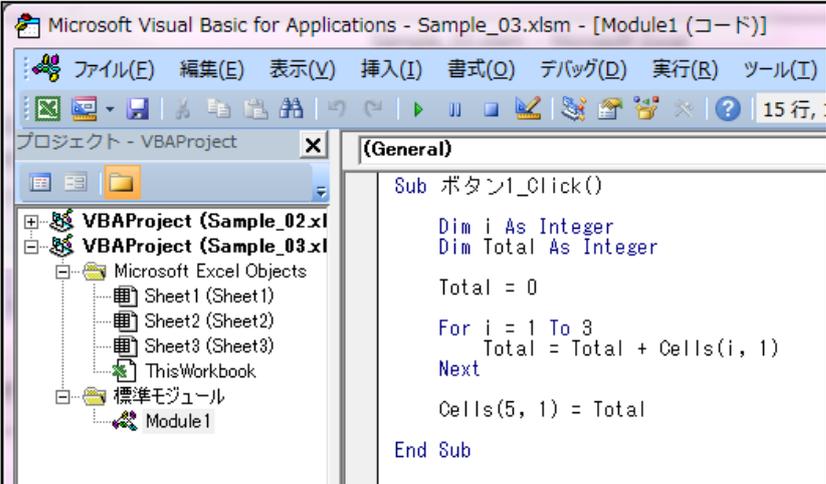
実行順序：
1 → 2 → 3 (Total = 0) → 4 (i ≦ 3 かどうか判断)
→ 5 (i = 1、Total = 0 + 「A1」)→ 4
→ 5 (i = 2、Total = 0 + 「A1」 + 「A2」) → 4
→ 5 (i = 3、Total = 0 + 「A1」 + 「A2」 + 「A3」) → 4
→ 6 → 7　を実行

補足：なぜ繰り返し処理を使用するのか？

上記の場合、Total = 「A1」 + 「A2」 + 「A3」 とすれば 1回の処理で済むのに、と感じるかもしれません。

繰り返し回数が多くなった場合や繰り返し終了がプログラムを開始した時点ではわからない場合などに繰り返し処理は威力を発揮します。

繰り返し処理を使用するプログラムの具体例です。
- セル A1 ～ A100 を加算してセル A101 に表示する、セル B1 ～ B900 の平均値を求める
- [ボタン1] を押すと測定器が測定し、測定値を受け取る動作繰り返し、[ボタン2] を押すと繰り返しを終了する

このプログラムは以下のプログラムで実現できます。

```
Sub ボタン1_Click()

    Dim i As Integer
    Dim Total As Integer

    Total = 0

    For i = 1 To 3
        Total = Total + Cells(i, 1)
    Next

    Cells(5, 1) = Total

End Sub
```

このプログラムでは　For i = 1 To 3 から Next までが繰り返しの部分です。
For Next の繰り返しでは、繰り返し回数をカウントする変数 (カウンタ変数、ここでは i) を指定します。
繰り返しを実行する毎に、カウンタ変数が初期値 (ここでは i = 1) から 1 ずつ加算され、To で指定した値 (ここでは 3) 以下であれば、繰り返し処理を実行します。

	A	B	C	D
1	3			
2	5		ボタン1	
3	1			
4				
5	9			
6				

具体的に、セル A1 には「3」、セル A2 には「5」、セル A3 には「1」というデータ (値) を入力し、[ボタン1] を押すとセル A1 から順番にセルの数値を加算し、結果「9」をセル A5 に表示する動作について説明します。

このプログラムでは変数 Total を使用することにします。
変数 Total は初期値は「0」で格納しているデータとセル A1、A2、A3 のデータを順番に加算し、結果を格納します。

Total = Total + Cells(1, 1) ⇨ Total (3 が代入される) ← 0 + 3 (セル A1 のデータ)
Total = Total + Cells(2, 1) ⇨ Total (8 が代入される) ← 3 + 5 (セル A2 のデータ)
Total = Total + Cells(3, 1) ⇨ Total (9 が代入される) ← 8 + 1 (セル A3 のデータ)
を実行します。
Total = Total + Cells(1, 1) というプログラムは、数学の式とは異なり、正しいプログラムです。

このような書き方は For Next などの繰り返しと合わせて何らかの数値の合計を求めるために、よく使用する方法です。

補足：インデント (字下げ)

VBA では Sub に対して End Sub 、For に対して Next など処理の開始と終了の両端を示すプログラムを記述します。
プログラムの構造をわかりやすくするために、**インデント (字下げ)** を使用することがあります。
インデントは半角スペース 2 個分であったり、キーボードの Tab キーを 1 回押すなどで字下げをします。
ここでは Tab キーを 1 回押すことでインデントを使用しています。

インデントなしの例：

```
For i = 1 To 10
Total = Total + Cells(i, 1)
Next
```

インデント

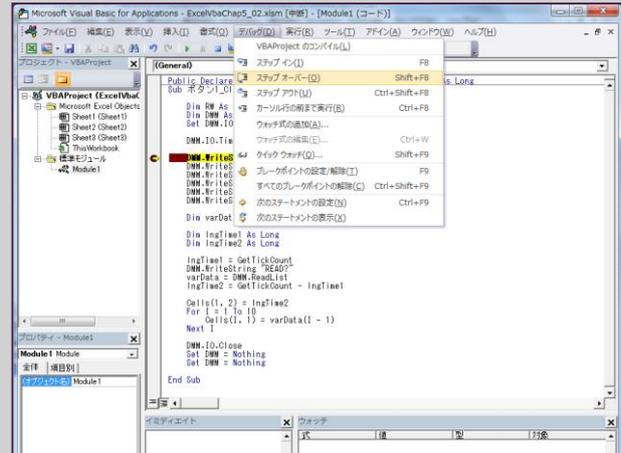
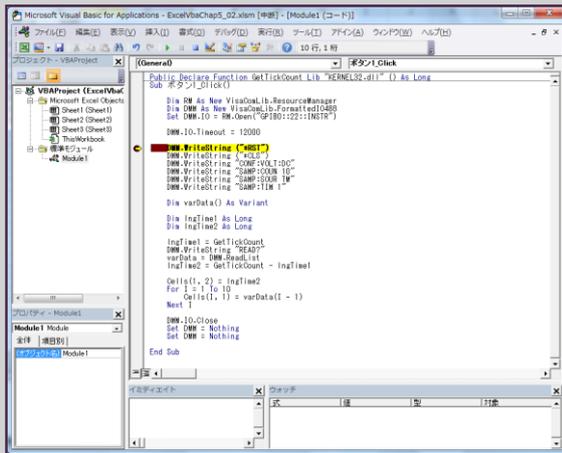
インデントありの例：

```
For i = 1 To 10
    Total = Total + Cells(i, 1)
Next
```

補足：ブレークポイントとステップ実行

プログラムがどの順番で動作しているかや、エラーが発生した場合どこでエラーが発生しているかを確認するにはブレークポイントとステップ実行を使用します。

– ブレークポイント
  プログラムの左側をクリックすると　●　が表示され、その行にブレークポイントが設定されます。
  プログラムの実行を開始すると、プログラムはブレークポイントの行を実行する前に、一時停止します。

– ステップ実行
  一時停止している状態から、Visual Basic Editor のメニューから、デバッグ → ステップオーバー　を実行すると、プログラムが 1行だけ実行されます。

条件分岐処理や繰り返し処理がどのように動作するか、測定器がどこでエラーを発生させたのかを確認するには、上記の方法で、ブレークポイントとステップ実行を使用し、一つずつコマンドを送り、どのコマンドでエラーが発生したのかを確認します。

4. 測定器制御の概要

Excel / VBA のプログラムで測定器を制御する場合、プログラムでは 「測定器にコマンドを送り、応答を受け取る」という処理を行います。
ここでは、Excel / VBA を使用して測定器にコマンドを送り、応答を受け取る方法を説明します。

補足：測定器のマニュアルで確認すべきこと

測定器に送るコマンドと返される応答は、使用する測定器毎に異なりますので、プログラムを作成するには
– どのようなコマンドがサポートされているか？
– どのような応答が返されるか？
の情報を、使用する測定器のマニュアルで確認する必要があります。

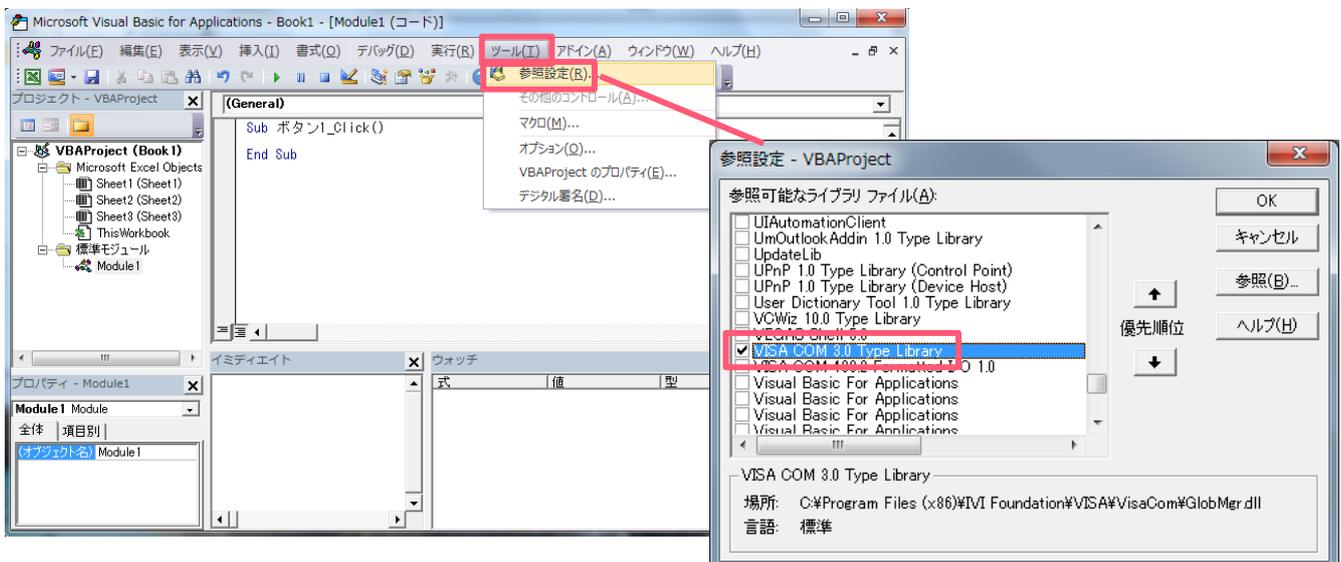
標準の Windows や Excel / VBA には測定器にコマンドを出力したり、測定器の応答を受け取ったりするためのライブラリは含まれていません。
キーサイトでは Excel / VBA などのプログラム開発環境に、測定器へコマンドを送ったり、測定器から応答を受け取る機能を提供するために、IO Libraries Suite というソフトウェアを提供しています。
IO Libraries Suite を PC にインストールすると、VISA COM ライブラリも同時にインストールされます。
VISA COM ライブラリを Excel / VBA から呼び出すことにより、PC に接続された測定器にコマンドを送ったり、測定器からの応答を受け取ったりすることができます。

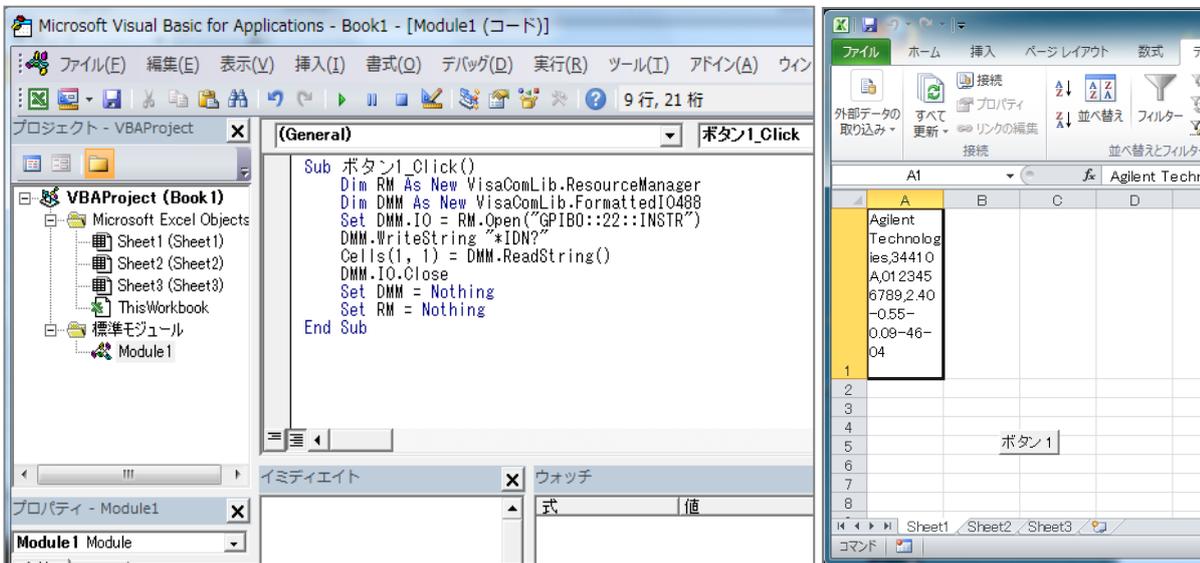
[IO Libraries Suite 簡易取扱説明書 最新版]
http://www.keysight.co.jp/find/iolib_jp_quick

5. 測定器制御の例

測定器にコマンドを文字列として送信し、応答を文字列として受け取る例を示します。
具体的な手順やプログラムの詳細は下記アプリケーション・ノートをご参照ください。

[Excel 2010 / VBA による測定器制御プログラム入門 VISA COM ライブラリ編]
http://literature.cdn.keysight.com/litweb/pdf/5991-2363JAJP.pdf
または「5991-2363JAJP」 で web 検索してください。

1. ボタンを作成
2. ツール → 参照設定で VISA COM ライブラリを追加

3.　プログラムの作成

```
Sub ボタン1_Click()

    Dim RM As New VisaComLib.ResourceManager
    Dim DMM As New VisaComLib.FormattedIO488
    Set DMM.IO = RM.Open("GPIB0::22::INSTR")

    DMM.WriteString "*IDN?"
    Cells(1, 1) = DMM.ReadString()

    DMM.IO.Close
    Set DMM = Nothing
    Set RM = Nothing

End Sub
```

補足：測定器の VISA アドレス

プログラムで指定する測定器のアドレスは VISA アドレスで指定します。
PC に接続されている測定器の VISA アドレスは IO Libraries の Connection Expert でご確認ください。

RS-232C の例："ASRL1::INSTR"
　→　COM1 に接続されている測定器

GPIB の例： "GPIB0::22::INSTR"
　→　GPIB0 のインタフェースに接続されている GPIB アドレス 22番の測定器

LAN の例 1："TCPIP0::192.168.1.1::inst0::INSTR“
　→　VXI-11 という測定器制御用プロトコルで接続された IP アドレス 192.168.1.1 の測定器

LAN の例 2："TCPIP0::192.168.1.1::5025::SOCKET“
　→　TCP Socket プロトコルでポート番号 5025 で接続された IP アドレス 192.168.1.1 の測定器

LAN の例 3："TCPIP0::192.168.1.1::hislip0::INSTR“
　→　HiSLIP という測定器制御用プロトコルで接続された IP アドレス 192.168.1.1 の測定器

USB の例："USB0::0x0957::0x2007::MY49000104::0::INSTR“
　→　USB で接続された測定器（個体毎に VISA アドレスが異なります）

参考資料

- IO Libraries Suite の簡易取扱説明書

[IO Libraries Suite 簡易取扱説明書 最新版]
http://www.keysight.co.jp/find/iolib_jp_quick
2015年6月時点ではバージョン 17の簡易取扱説明書にリンクしています。

[IO Libraries Suite 16.2/16.3 簡易取扱説明書] （5991-0331JAJP）
http://literature.cdn.keysight.com/litweb/pdf/5991-0331JAJP.pdf

- プログラミング関連資料

[Excel 2010 / VBA による測定器制御プログラム入門 VISA COM ライブラリ編] （5991-2363JAJP）
http://literature.cdn.keysight.com/litweb/pdf/5991-2363JAJP.pdf

[Excel / VBA による測定器制御プログラム入門 中級編] （5992-0690JAJP）
http://literature.cdn.keysight.com/litweb/pdf/5992-0690JAJP.pdf

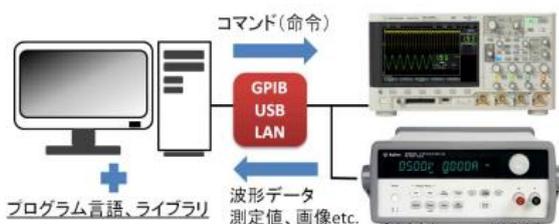
[自動計測プログラム WEB]
http://www.keysight.co.jp/find/sample_program
測定器のサンプル・プログラムなどを提供しております

[これまでにリリースされたシステム開発ガイド一覧]
http://www.keysight.com/main/editorial.jspx?cc=JP&lc=jpn&ckey=687300-3-jpn&id=687300-3-jpn

[Test-System Development Guide] （5989-5367EN）
http://literature.cdn.keysight.com/litweb/pdf/5989-5367EN.pdf

[Using VISA COM I/O API in .NET] （5989-6338EN）
http://literature.cdn.keysight.com/litweb/pdf/5989-6338EN.pdf

- 有償セミナのご案内

[自動制御トレーニング・コース]
http://www.keysight.co.jp/find/programing103

キーサイト･テクノロジー合同会社
本社〒192-8550 東京都八王子市高倉町9-1

計測お客様窓口
受付時間9:00-18:00（土・日・祭日を除く）
TEL 0120-421-345 (042-656-7832)
FAX 0120-421-678 (042-656-7840)
Email contact_japan@keysight.com
ホームページ www.keysight.co.jp

記載事項は変更になる場合があります。
ご発注の際はご確認ください。

Published in Japan, June 12, 2015
5992-0691JAJP
0000-08A